

# Exhibit 8

**IN THE UNITED STATES DISTRICT COURT  
FOR THE SOUTHERN DISTRICT OF NEW YORK**

<b>DROPLETS, INC.,</b>	)	
	)	<b>Civil Action No. 1:12-cv-02326-CM</b>
<b>Plaintiff,</b>	)	
	)	<b>ECF CASE</b>
<b>v.</b>	)	
	)	
<b>E*TRADE FINANCIAL</b>	)	
<b>CORPORATION, et al.,</b>	)	
	)	
<b>Defendants.</b>	)	
	)	
	)	
	)	

---

**MEMORANDUM OF LAW IN SUPPORT OF DEFENDANTS' MOTION FOR  
SUMMARY JUDGMENT OF NON-INFRINGEMENT**

## TABLE OF CONTENTS

	<u>Page</u>
TABLE OF AUTHORITIES.....	iii
I. INTRODUCTION.....	1
II. BACKGROUND.....	1
III. UNDISPUTED FACTS .....	2
A. The Four Representative Products .....	2
1. Interactive Charting.....	3
2. Screener.....	4
3. Dock .....	5
4. Search Suggest .....	6
B. User-Agent Headers .....	7
IV. LEGAL STANDARDS .....	8
A. Summary Judgment.....	8
B. Infringement .....	8
V. ARGUMENT .....	9
A. Under The Court’s Claim Construction, None Of The Four Accused Sets Of JavaScript Code Are An Interactive Link .....	9
1. The accused JavaScript cannot be the interactive link because it is not “selected by an end user” .....	10
a. For Interactive Charting, chart.js is not selected by an end user .....	11
b. For Screener, ScreenerUtility.js.package.js is not selected by an end user.....	12
c. For Dock, tda.js is not selected by an end user .....	13
d. For Search Suggest, jquery.autocomplete.ext.js is not selected by an end user .....	14
2. The accused JavaScript cannot be the interactive link because it lacks “facilities for restoring previous operating states of the application” .....	15
a. For all four representative products, the accused JavaScript is not personalized for any user and therefore cannot restore a user’s prior operating state .....	16
b. With respect to Search Suggest, no state is restored at all .....	18
c. For Claim 26, the accused JavaScript cannot be the interactive link because the accused JavaScript cannot restore itself .....	18

3.	The accused JavaScript cannot be the interactive link because it is “instructions to perform on an Internet browser” .....	19
B.	Under the Court’s Claim Construction, The Four Representative Products Do Not Send “Operating Environment Information” .....	20
1.	The four representative products do not send hardware information as required by Claim 1 .....	20
2.	Claim 1’s “operating environment information” limitation applies equally to Claim 26 .....	23
C.	Because Droplets Cannot Identify Any “Interactive Link” Or “Operating Environment” Information, There Is No Infringement .....	24
VI.	CONCLUSION .....	25

## TABLE OF AUTHORITIES

Page(s)

## CASES

<i>Anderson v. Liberty Lobby Inc.</i> , 477 U.S. 242 (1986) .....	8
<i>Celotex Corp. v. Catrett</i> , 477 U.S. 317 (1986) .....	8
<i>Cross Med. Prods., Inc. v. Medtronic Sofamor Danek</i> , 424 F.3d 1293 (Fed. Cir. 2005) .....	8
<i>DSU Med. Corp. v. JMS Co., Ltd.</i> , 471 F.3d 1293 (Fed. Cir. 2006) .....	9, 24
<i>Dynacore Holdings Corp. v. U.S. Phillips Corp.</i> , 363 F.3d 1263 (Fed. Cir. 2004) .....	24
<i>Johnston v. IVAC Corp.</i> , 885 F.2d 1574 (Fed. Cir. 1989) .....	9, 24
<i>Limelight Networks, Inc. v. Akamai Techs., Inc.</i> , 134 S. Ct. 2111 (2014) .....	9
<i>Network Commerce, Inc. v. Microsoft Corp.</i> , 422 F.3d 1353 (Fed. Cir. 2005) .....	9, 24
<i>Novartis Corp. v. Ben Venue Labs., Inc.</i> , 271 F.3d 1043 (Fed. Cir. 2001) .....	8
<i>Pickholtz v. Rainbow Techs., Inc.</i> , 284 F.3d 1365 (Fed. Cir. 2002) .....	22
<i>PSN Ill. LLC v. Ivolcar Vivadent, Inc.</i> , 525 F.3d 1159 (Fed. Cir. 2008) .....	8
<i>Southwall Tech. Inc. v. Cardinal IG Co.</i> , 54 F.3d 1570 (Fed. Cir. 1995) .....	23
<i>Taurus IP, LLC v. Daimler Chrysler Corp.</i> , 726 F.3d 1306 (Fed. Cir. 2013) .....	8, 9
<i>Under Sea Indus. v. Dacor Corp.</i> , 833 F.2d 1551 (Fed. Cir. 1987) .....	8

### **TABLE OF ABBREVIATIONS**

“the ’745 Patent”	U.S. Patent No. 6,687,745
“the ’838 Patent”	U.S. Patent No. 7,502,838
“the ’115 Patent”	U.S. Patent No. 8,402,115
“the accused JavaScript”	The sets of four JavaScript code Droplets accuses of being an “interactive link”: chart.js for the Interactive Charting representative product; jquery.autocomplete.ext.js for Search Suggest; ScreenerUtility.js.package.js for Screener; and tda.js for Dock.
Claims 1 and 26	Claims 1 and 26 of U.S. Patent No. 6,687,745
“Defendants”	Collectively refers to all Defendants in this litigation
“Ex.”	Refers to the corresponding Exhibit of the Declaration of Brian D. Range submitted with this brief
“representative products”	Collectively refers to all four representative products: TD Ameritrade’s Interactive Charting, E*TRADE’s Search Suggest, E*TRADE’s Screener, and TD Ameritrade’s Dock. <sup>1</sup>
“E*TRADE”	Refers to all E*TRADE defendants in this litigation
“Scottrade”	Refers to all Scottrade defendants in this litigation
“TD Ameritrade”	Refers to all TD Ameritrade defendants in this litigation

---

<sup>1</sup> The parties have used both “Search” and “Search Suggest” to refer to the accused search functionality. Also, note that Defendants use the word “product” in this brief as a patent term of art. The representative products are, in reality, web site features.

## INDEX OF DECLARATIONS AND EXHIBITS

### **Submitted with Defendants' Motion for Summary Judgment**

Declaration of Brian D. Range with the following Exhibits:

- Exhibit 1: Letter from Michael B. Levin to Josh Budwin dated February 5, 2014
- Exhibit 2: Declaration of Michael I. Shamos, Ph.D., dated July 2, 2014
- Exhibit 3: Excerpted transcript of the deposition of David Martin taken July 10, 2014
- Exhibit 4: Video entitled "Interactive Charting.mov" including blurring to obscure potentially confidential information
- Exhibit 5: Declaration of David Martin, Ph.D. dated June 13, 2014
- Exhibit 6: Video entitled "Screener.mov" including blurring to obscure potentially confidential information
- Exhibit 7: Video entitled "Dock.mov" including blurring to obscure potentially confidential information
- Exhibit 8: Video entitled "Search Suggest.mov" including blurring to obscure potentially confidential information
- Exhibit 9: Excerpted transcript of the deposition of Michael Shamos taken July 14, 2014
- Exhibit 10: U.S. Patent No. 6,687,745 ("745 Patent") issued on February 3, 2004 with the Inter Partes Reexamination Certificate issued March 1, 2011
- Exhibit 11: July 2, 2009 Amendment to the Claims and Remarks excerpted from the U.S. Patent & Trademark Office Reexamination file history for the '745 Patent
- Exhibit 12: November 17, 2008 Amendment to the Claims and Remarks excerpted from the U.S. Patent & Trademark Office Reexamination file history for the '745 Patent
- Exhibit 13: August 12, 2003 Amendment and Response to Office Action excerpted from the U.S. Patent & Trademark Office Prosecution file history for the '745 Patent

## I. INTRODUCTION

Trying to make sense of Droplets’ infringement theories is like trying to fit a square peg into a round hole. After (correctly) dropping its original infringement theory—that ordinary browser bookmarks constitute the “interactive link”—Droplets has manufactured a new (but equally meritless) theory—that certain JavaScript code is the “interactive link” as claimed. It is undisputed, however, that the accused JavaScript is executed only if a user navigates to a particular web page via hyperlinks and, rather than including operating state information, only contains generic web browser instructions for contacting a server. As a result, the accused JavaScript is not “selected by an end user” and does not include “facilities for restoring” previous operating states of a user’s application, both of which are required under this Court’s construction of “interactive link.” Similarly, Droplets’ infringement theory with respect to the “information relating to the operating environment of the client computer” claim element effectively eliminates one of three requirements from this Court’s construction (specifically, the “hardware capabilities” requirement).

When the Court’s constructions for “interactive link” and “information relating to the operating environment of the client computer” are properly interpreted and applied, the undisputed facts confirm that the four representative products accused by Droplets fail to include these elements. Droplets cannot make a square peg fit a round hole. Summary judgment of non-infringement is appropriate.

## II. BACKGROUND

Defendants E\*TRADE, Scottrade, and TD Ameritrade are leading financial services companies who offer information to customers and prospective customers through their web sites. Droplets alleges that Defendants’ web sites infringe three U.S. Patents: the ’745 Patent, the ’838 Patent, and the ’115 Patent. Dkt. 248. The Court stayed litigation as to the ’838 Patent and ’115 Patent pending post-grant patent review by the U.S. Patent and Trademark Office. *See* Dkt. 218 at 3; Dkt. 245 at 3. Thus, only the ’745 Patent is now at issue. The instant motion, if granted, would eliminate all ’745 Patent infringement issues from this litigation.



Through several rounds of briefing and a *Markman* hearing, this Court issued two claim construction rulings. Dkt. 218 and 242. Shortly after this Court's second claim construction ruling, Defendants sent Droplets a letter explaining why Droplets could not properly allege infringement in light of the Court's claim constructions. Ex. 1. Despite the Court's constructions, Droplets declined to abandon its claims of infringement.

The parties agreed, however, that judicial efficiency would best be served by stipulating to a streamlined summary judgment procedure so the Court could efficiently determine whether Droplets has any viable infringement claims. Dkt. 257. Under the stipulated procedure, Defendants permitted Droplets to revise its infringement contentions. *Id.* at ¶ 1. Further, the parties agreed that the summary judgment procedure would address only limited non-infringement issues (*id.* at ¶ 5) and that no fact discovery is necessary to resolve these limited issues (*id.* at ¶ 6). The Court approved the stipulation on May 20, 2014. Dkt. 257.

In accordance with that stipulated procedure, Droplets has provided new infringement contentions, the parties have exchanged expert declarations on the “interactive link” and “operating environment” infringement issues, and the parties deposed each other's experts on the subject matter of those declarations.<sup>2</sup>

### III. UNDISPUTED FACTS

#### A. The Four Representative Products

In its revised infringement contentions, Droplets identified four “representative products” and explained a new theory for how each representative product allegedly infringes the only asserted independent claims of the '745 Patent, Claims 1 and 26. The four representative products are: interactive charting (as represented by TD Ameritrade's system); screener (as represented by E\*TRADE's system); TD Ameritrade's dock; and search suggest (as represented by E\*TRADE's

---

<sup>2</sup> Defendants' products do not infringe the '745 Patent for many reasons beyond those raised in this motion. Defendants have tried to present in this motion some of the more straightforward reasons why there is no infringement and do so to foreclose an argument by Droplets that fact discovery is necessary before the issues can be resolved. Defendants reserve all rights to raise other bases of non-infringement to the extent further proceedings are necessary.

system). The parties agreed that summary judgment of non-infringement as to any representative product also applies to Defendant products that are not colorably different. Dkt. 257 at ¶¶ 3-4. Thus, resolution of the infringement issues as to the representative products should resolve infringement issues for all of Defendants' products. *Id.* at ¶ 4.

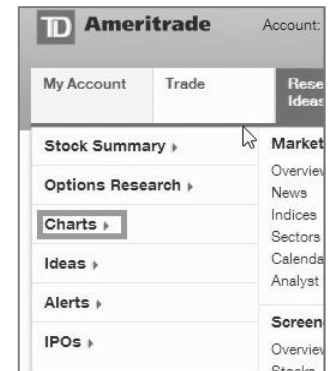
At a high level, the four representative products have similar features that result in the alleged infringement: each product is a website feature that allows users to interact with information and data; each is accessed by a user navigating to a web page; each uses JavaScript code to provide some functionality. Also, for each of the four representative products, Droplets alleges that corresponding JavaScript code is the "interactive link." JavaScript code consists of instructions to a web browser. Ex. 2 at ¶¶ 60, 88, 110, 123; Ex. 3 at 64:6-9.

The experts do not dispute either the operation of the four representative products or how each product utilizes the accused JavaScript code. Below, Defendants explain the undisputed operation of each representative product, as demonstrated in videos produced by Droplets' expert, Dr. Martin, and as explained by Dr. Martin during his deposition.<sup>3</sup>

### 1. Interactive Charting

TD Ameritrade's Interactive Charting accused product allows a user to see price history for a security or investment index in chart form. Ex. 5 at ¶ 26. Dr. Martin's "Charting.mov" video depicts an allegedly infringing use of Interactive Charting. Ex. 4; Ex. 3 at 38:7-18.

A user may view interactive charts by opening a browser and navigating to the TD Ameritrade web site home page, entering a username and password, and pressing "enter" on the keyboard or clicking the on-screen Log On button. The user then hovers over or clicks on the Research & Ideas tab and, finally, clicks on the Charts hyperlink. *Id.* at 40:6-41:10; 41:20-42:3; 42:21-43:2;



<sup>3</sup> For purposes of this motion, Defendants rely on certain facts presented by Droplets' expert, Dr. Martin, without questioning their accuracy. Defendants rely on these facts solely for purposes of this motion, and reserve all rights to later contest such alleged facts if further proceedings become necessary.

44:25-45:3. Selecting the Charts hyperlink returns a web page at a different URL. *Id.* at 44:10-24. The user can change the displayed security (in the video, changing from MSFT to CSCO) and change the displayed date range. *Id.* at 50:2-51:13. If the user logs off, closes the browser, reopens the browser, logs back on, and once again selects the Charts hyperlink, a chart with the most recently chosen security (in the video, CSCO) and most recently chosen date range is again displayed. *Id.* at 59:17-61:15.



The alleged interactive link for Interactive Charting is the chart.js JavaScript code. The client computer requests, retrieves, and executes the chart.js JavaScript code from the server after the user clicks on the Charts hyperlink. *Id.* at 65:1-66:7; Ex. 2 at ¶¶ 56-57.

The chart.js code is a set of browser instructions that asks the server to send a chart. Chart.js has no knowledge of any previously displayed chart and contains no operating state information. Ex. 9 at 53:15-54:21; Ex. 3 at 66:8-12. All copies of the same release of chart.js are exactly the same on every computer to which they have been downloaded. *Id.* Because any personalization is saved at the server instead of within chart.js, the Charts feature can restore some of the user's customization even when the user logs on to his or her account from a different computer (without transferring a user-customized interactive link). Ex. 2 at ¶ 50.

## 2. Screener

E\*TRADE's Screener product allows a user to narrow a large set of exchange-traded fund ("ETF") choices into a smaller set according to criteria such as those designated All Star, those that are commission-free, those having a certain yield, and so forth. Ex. 5 at ¶ 142. Dr.

ETF Name (symbol)	Fund Category	Morningstar Rating	Market Price	Expense Ratio	Premium/Discount	Liquidity Factor
iShares 10-Year Credit Bond ETF (CLST)	Corporate Bond	***	\$99.00	0.20%	0.15%	...
iShares 10-Year High Yield Corporate Bond ETF (HYG)	High Yield Bond	***	\$94.49	0.50%	0.43%	...
iShares 10-Year Investment Grade Corporate Bond ETF (IGSB)	Corporate Bond	***	\$118.28	0.15%	0.17%	...
iShares J.P. Morgan USD Emerging Markets Bond ETF (EMBD)	Emerging Markets Bond	***	\$114.54	0.80%	0.61%	...
iShares MSCI EAFE Index Fund (EFA)	Emerging Markets Bond	***	\$24.16	0.44%	-0.84%	...
iShares MSCI USA Index Fund (IUSA)	High Yield Bond	*****	\$92.77	1.25%	-0.68%	...

Martin's "Screener.mov" video depicts allegedly infringing use of the Screener. Ex. 6; Ex. 3 at 38:7-18.

A user may view the Screener by opening a browser, navigating to the E\*TRADE home page, entering a username and password and clicking the Log On button. After logging, on the user is presented a web page having a different URL. The user then hovers over or clicks on the Research tab, clicks on the ETFs button, and finally clicks on the ETF Screener tab to navigate to a different web page at a different URL. *Id.* at 104:7-106:9; 108:4-10. Once on the Screener page, the user can then click on one of the Predefined Strategies to select ETFs based on a collection of strategies defined in advance by E\*TRADE, such as "Top-Performing Domestic Equity ETFs." *Id.* at 109:2-110:10. If the user then logs out, closes the browser, reopens the browser, logs back on, and navigates back to the Screener page, the user's most recent choice of filtered results is displayed. *Id.* at 115:9-116:15.

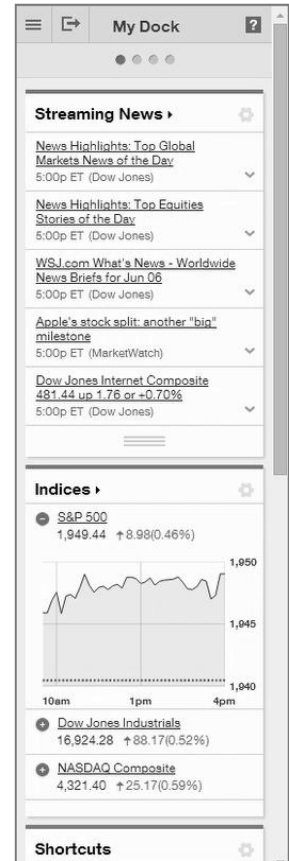
The alleged interactive link for the Screener is the ScreenerUtility.js.package.js JavaScript code. The client computer requests and retrieves the ScreenerUtility.js.package.js JavaScript when the user clicks on the ETF Screener tab. *Id.* at 117:13-118:4. The ScreenerUtility.js.package.js code consists of browser instructions that, after being retrieved from the server and executed on the client, ask the server to send ETF choices. ScreenerUtility.js.package.js contains no operating state information. *Id.* at 118:11-19. All copies of the same release of ScreenerUtility.js.package.js are exactly the same on every computer to which they have been downloaded. Because any personalization is saved at the server instead of within ScreenerUtility.js.package.js, the Screener feature can provide the same results on different computers (without transferring a user-customized interactive link). Ex. 2 at ¶ 50.

### **3. Dock**

TD Ameritrade's accused Dock functionality allows users to select and arrange different modules (such as Streaming News, Indices, and Shortcuts) to be displayed to the user once the user logs on to the site. Ex. 5 at ¶ 198. Dr. Martin's "Dock.mov" video depicts allegedly infringing use of the Dock. Ex. 7; Ex. 3 at 38:7-18.

A user may view the Dock by opening a browser, navigating to the TD Ameritrade web site, entering a username and password, and pressing “enter” on the keyboard or clicking the on-screen Log On button. *Id.* at 41:17-19; 69:21-70:21; 75:10-15; Ex. 7 at Dock.mov. After logging on, the user is presented a web page with a new URL where the Dock is accessible. Users may click and “drag” modules within the Dock, for example the Streaming News module, to appear at a different location within the Dock. Ex. 3 at 70:25-71:8. If the user logs out, closes the browser, opens the browser, and logs back on, the Dock is displayed with the modules in the same general arrangement as the user left them. *Id.* at 74:17-75:25.

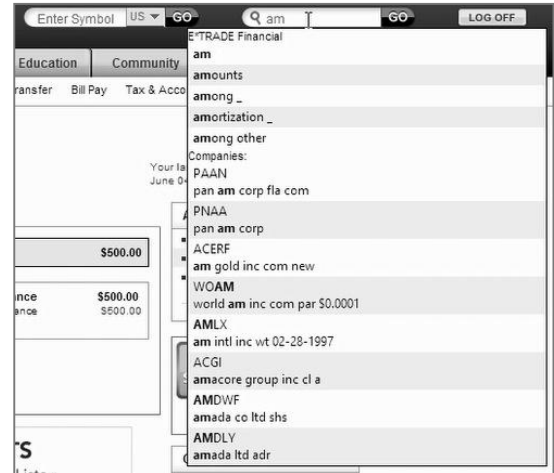
The alleged interactive link for the Dock is a file of JavaScript code named tda.js. Dr. Martin testified that he did not investigate when the client computer requests and retrieves the tda.js JavaScript. *Id.* at 77:20-78:9. In his declaration, however, he suggests that the tda.js JavaScript code is retrieved when the user “navigates to the dock” (in other words, by logging on to the TD Ameritrade web site). Ex. 5 at ¶ 203. The tda.js code consists of browser instructions that, after being retrieved from the server and executed on the client, ask the server to provide instructions for displaying the dock. No operating state information is stored within the tda.js JavaScript code. Ex. 3 at 78:20-25. All copies of the same release of tda.js are exactly the same on every computer to which they have been downloaded. Because any personalization is saved at the server instead of within tda.js, the Dock feature can restore some of the user’s customization even when the user logs on to his or her account from a different computer (without transferring a user-customized interactive link). Ex. 2 at ¶ 117-18.



#### 4. Search Suggest

E\*TRADE’s web site provides an industry-standard “search suggest” feature. Ex. 5 at ¶ 87, Fig. 11. Dr. Martin’s “Search suggest.mov” video depicts an allegedly infringing use of the Search Suggest product. Ex. 8; Ex. 3 at 38:7-18.

A user may access the Search Suggest product by opening a browser, navigating to the E\*TRADE home page, entering username and password, and pressing “enter” or the Log On button. *Id.* at 82:2-84-1. After logging on, the user is presented a web page that includes the Search Suggest product. Ex. 8. If the user then types at least two letters into the search box, different possible phrase completions appear below. Ex. 3 at 84:3-20. The



user can select a highlighted word (such as “amortization” in the video), to navigate to a different web page having a different URL that displays search results based on the chosen completion. *Id.* at 92:12-16; 93:1-17; 95:4-7.

The alleged interactive link for Search Suggest is the jquery.autocomplete.ext.js JavaScript code. The client computer requests and retrieves the jquery.autocomplete.ext.js JavaScript when a user navigates to a web page containing the search application (in other words, by logging on to the E\*TRADE site). Ex. 5 at ¶ 92; Ex. 3 at 98:2-21. The jquery.autocomplete.ext.js code is a set of browser instructions that collects characters typed into the search box and sends them to the server. Ex. 2 at ¶ 88. Search Suggest does not provide for any user customization options. *See* Ex. 8; Ex. 2 at ¶¶ 82-84.

## B. User-Agent Headers

Droplets alleges the “operating environment” information limitation of the ’745 Patent claims is satisfied by a user-agent header. The user-agent header is a string sent with a hypertext transfer protocol (HTTP) request that conveys information to a server about the client program making the request. Ex. 5 at ¶¶ 40-41.

User-agent headers may be used in conjunction with each representative product. *Id.* at ¶ 39. Droplets identifies one particular user-agent header as used with representative products: “User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36.” *Id.* at ¶ 41. Microsoft defines WOW64 as “the



x86 emulator that allows 32-bit Windows-based applications to run seamlessly on 64-bit Windows.” *Id.* at ¶ 42(c). Both sides’ experts expect that some (but not all) information about hardware can always be inferred from information about operating system or browser. Ex. 3 at 132:5-10; *id.* at 129:11-132:4; Ex. 9 at 121:10-123:12; Ex. 2 at ¶ 75.

#### IV. LEGAL STANDARDS

##### A. Summary Judgment

Summary judgment is appropriate where the non-moving party fails to make a showing sufficient to establish the existence of an element essential to its case and on which it will bear the burden of proof at trial. *Novartis Corp. v. Ben Venue Labs., Inc.*, 271 F.3d 1043, 1046 (Fed. Cir. 2001); *see also Celotex Corp. v. Catrett*, 477 U.S. 317, 325 (1986); *Taurus IP, LLC v. Daimler Chrysler Corp.*, 726 F.3d 1306, 1325-26 (Fed. Cir. 2013) (affirming summary judgment of non-infringement where “Taurus identified no evidence web surfers can create and edit rules that ‘govern the relationship’ between data items, as required by the district court’s claim construction”); *PSN Ill. LLC v. Ivolcar Vivadent, Inc.*, 525 F.3d 1159, 1166-67 (Fed. Cir. 2008) (affirming summary judgment of non-infringement where accused features were not “ready for mounting” as construed by the court).

In assessing whether any genuine issue of material fact precludes summary judgment, not every disputed fact is deemed material for the purposes of the analysis: “Only disputes over facts that might affect the outcome of the suit under the governing law will properly preclude summary judgment.” *Anderson v. Liberty Lobby Inc.*, 477 U.S. 242, 248 (1986).

##### B. Infringement

Droplets bears the burden of proving that Defendants infringe the ’745 Patent. *See Under Sea Indus. v. Dacor Corp.*, 833 F.2d 1551, 1557 (Fed. Cir. 1987) (“The burden is always on the patentee to show infringement.”). “To prove direct infringement, the plaintiff must establish by a preponderance of the evidence that one or more claims of the patent read on the accused device literally or under the doctrine of equivalents.” *Cross Med. Prods., Inc. v. Medtronic Sofamor*

*Danek*, 424 F.3d 1293, 1310 (Fed. Cir. 2005) (citation omitted). To establish literal infringement, “every limitation set forth in a claim must be found in an accused product or process exactly.” *Johnston v. IVAC Corp.*, 885 F.2d 1574, 1577-78 (Fed. Cir. 1989). If the accused product lacks a claim limitation, summary judgment of non-infringement is proper. *See, e.g., Taurus IP, LLC*, 726 F.3d at 1325-26.

To establish infringement under the doctrine of equivalents, the patentee must “provide particularized testimony and linking argument as the insubstantiality of the differences between the claimed invention and the accused device or process, or with respect to the function, way, result test when such evidence is presented to support a finding of infringement under the doctrine of equivalents. Such evidence must be presented on a limitation-by-limitation basis.” *Network Commerce, Inc. v. Microsoft Corp.*, 422 F.3d 1353, 1363 (Fed. Cir. 2005).

Droplets also alleges that Defendants indirectly infringe the ’745 Patent by encouraging end users to practice the patent by using Defendants’ accused products. Dkt. 248. Absent direct infringement, there can be no indirect infringement. *Limelight Networks, Inc. v. Akamai Techs., Inc.*, 134 S. Ct. 2111, 2115 (2014) (holding that defendant may not be liable for inducing patent infringement when no one has directly infringed the patent); *DSU Med. Corp. v. JMS Co., Ltd.*, 471 F.3d 1293, 1303 (Fed. Cir. 2006) (en banc) (“[T]he patentee always has the burden of showing direct infringement for each instance of indirect infringement.”).

## V. ARGUMENT

### A. Under The Court’s Claim Construction, None Of The Four Accused Sets Of JavaScript Code Are An Interactive Link

Claims 1 and 26 (the only two asserted independent claims) each require an “interactive link.” To obtain its patent despite the very crowded field of prior art, Droplets argued before the USPTO that the “interactive link” was key to its invention. At the outset of this case, Droplets accused ordinary browser bookmarks of satisfying this critical claim element. *See* Dkt. 204 at 11. Then, on January 28, 2014, this Court construed “interactive link” as follows:



Computer code that (1) retrieves and presents applications and/or information stored at remote locations across the network when selected by an end user, and (2) includes facilities for restoring previous operating states of the application as the application is re-presented at a user's computer. An interactive link cannot be a bookmark, cookie, shortcut, hyperlink or Internet address (URL).

Dkt. 218 and 242. In response, Droplets manufactured a new (and equally meritless) infringement theory, such that Droplets now accuses certain JavaScript code as being the “interactive link.”

As explained below, there are at least three reasons why the accused JavaScript cannot be an interactive link under the Court's constructions: the JavaScript code (1) is not selected by the end user, (2) does not restore operating state, and (3) consists of disclaimed “instructions to perform on an Internet browser.” Each reason is independent of the others. If the Court agrees with *any* of these reasons, summary judgment of non-infringement should be granted based on lack of an interactive link.

**1. The accused JavaScript cannot be the interactive link because it is not “selected by an end user”**

Under this Court's claim construction, the interactive link, among other things, “retrieves and presents applications and/or information stored at remote locations across the network when selected by an end user.” Dkt. 218 at 11 (emphasis added). This Court further held that “selected,” in the context of the '745 Patent, relates to the user clicking on a visual embodiment of code (such as text, an icon, a button, or another graphical representation):

“Link” is another commonly used computer term; it refers to computer code that, when selected by an end user (generally by “clicking” on its visual embodiment with a mouse), connects the user to other bits of code. In common parlance, links are selected by an end user when he wishes to retrieve data from the Internet and present (display) that data at his own terminal.”

*Id.* at 9-10 (citing Ex. 10 at 2:50-55). The '745 Patent also explains “selection” consistently with this Court's claim construction:

- “Once the interactive link and the graphical representation are downloaded, the graphical representation may then be selected.” Ex. 10 at 5:31-33.
- “In accordance with the present invention, and as depicted in FIG. 5, selection of the interactive link invokes the droplet-enabled application and/or information, e.g., the selection of the graphical representation 320 invokes the Stock Watcher application 100. That is, selecting the representation 320 (link) triggers the droplet supporting the Stock Watcher application 100 to re-execute on the client computer.” *Id.* at 16:41-48.

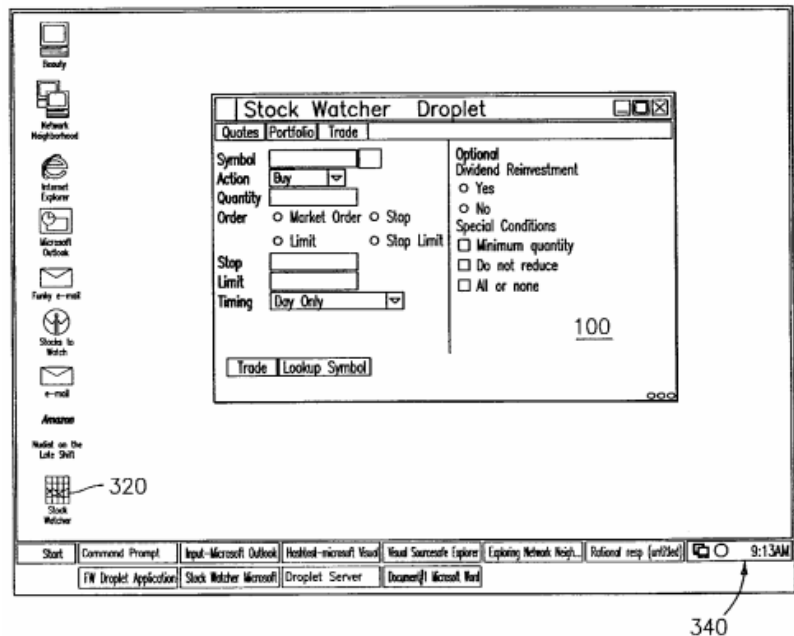


FIG. 5

See also, e.g., *id.* at 12:3-14 (referencing “selection (‘clicking’) of GUI components such as radio and command buttons”); *id.* at 15:24-16:2 (explaining selection of icons by clicking them with a pointing device). Thus, to “select” an interactive link means the user manually interacts with a user interface element (for example, by moving the mouse to select a graphical element and clicking on it) that visually embodies the interactive link code.

Droplets’ new infringement theory redefines what it means to “select” the interactive link in a manner inconsistent with the Court’s construction and the ’745 Patent. Specifically, Droplets takes the position that interactive link code may be “selected by an end user” if the user merely navigates to a web page, which in turn invokes or downloads the code. See, e.g., Ex. 5 at ¶¶ 37, 97, 152, 209. This theory cannot be correct, however, for the reasons outlined below.

**a. For Interactive Charting, chart.js is not selected by an end user**

Droplets presents no evidence that any user clicks on or otherwise selects a discrete visual embodiment or graphical representation of the chart.js JavaScript code (the purported “interactive link” for Interactive Charting) to trigger or invoke the Interactive Charting product. Instead,

Droplets alleges that the chart.js JavaScript code is “selected” by a user when the user navigates to the Charts web page by clicking on the Charts hyperlink. *Id.* at ¶ 37. There are at least three reasons why clicking on the Charts hyperlink is neither a selection of the charts.js code nor an interactive link selection.

First, “Charts” is a hyperlink. Ex. 3 at 47:3-13; *see also* Section III(A)(1), *supra*. Under the Court’s construction, the interactive link cannot be a hyperlink. Dkt. 242.

Second, the Charts hyperlink is not a visual embodiment or graphical representation of the chart.js JavaScript code. Instead, the Charts hyperlink is actually a visual embodiment of the location of a file containing *html* code, and clicking the Charts hyperlink causes that *html* code to fetch a web page via a URL from a remote server. Ex. 2 at ¶¶ 56-57.

Third, the client browser only requests and retrieves the chart.js JavaScript from the server *after* the client clicks on the Charts hyperlink to download the interactive charts web page. Ex. 3 at 65:1-66:7; Ex. 2 at ¶ 56. And the chart.js JavaScript code only executes *after* that web page is downloaded. *Id.* at ¶ 57. Thus, clicking the Charts hyperlink cannot constitute selection of the chart.js JavaScript code because that code is not even available to the client computer until after the Charts hyperlink is clicked.

Because charts.js does not meet the “when selected by an end user” requirement for an interactive link, the Charting product lacks any interactive link. Thus, Charts does not include all elements of Claims 1 or 26.

**b. For Screener, ScreenerUtility.js.package.js is not selected by an end user**

Droplets presents no evidence that any user clicks on or otherwise selects a discrete visual embodiment or graphical representation of the ScreenerUtility.js.package.js JavaScript code (the purported “interactive link” for Screener) to trigger or invoke the Screener product. Instead, Droplets alleges that JavaScript is “selected” by a user when the user navigates to the Screener web page by clicking on the ETF Screener tab. Ex. 5 at ¶ 152; Ex. 3 at 104:7-106:9. But, as with the

accused charts.js JavaScript code, there are at least three reasons why clicking on the ETF Screener tab is neither a selection of the accused JavaScript nor an interactive link selection.

First, the ETF Screener tab is a hyperlink. *Id.* at 107:14-108:15; *see also* Section III(A)(2), *supra*. The interactive link cannot be a hyperlink. Dkt. 242.

Second, the ETF Screener tab is not a visual embodiment or graphical representation of the ScreenerUtility.js.package.js JavaScript code. Instead, the ETF Screener tab is actually a visual embodiment of *html* code, not of the ScreenerUtility.js.package.js JavaScript code. Clicking on the ETF Screener tab merely connects the user to a different URL. Ex. 3 at 108:4-10.

Third, the client browser only requests and retrieves the ScreenerUtility.js.package.js JavaScript code from the server *after* the client clicks on the ETF screener tab. *Id.* at 117:13-24. Therefore, clicking the ETF Screener tab cannot constitute selection of the ScreenerUtility.js.package.js JavaScript code because that code is not even available to the client computer until after the user clicks on the ETF Screener tab.

Because ScreenerUtility.js.package.js does not meet the “when selected by an end user” requirement for an interactive link, the Screener product lacks any interactive link. Thus, the accused Screener product does not include all elements of Claims 1 or 26.

**c. For Dock, tda.js is not selected by an end user**

Rather than identifying any alleged visual embodiment or graphical representation of the tda.js JavaScript code (the purported “interactive link” for the Dock product), Droplets alleges that the code is “selected” by a user “when the user navigated to a web page containing the dock application” (i.e., when the user logs on to the TD Ameritrade web site). Ex. 5 at ¶¶ 209. Once again, there are three reasons why navigating to the Dock is not a selection of tda.js nor is it an interactive link selection.

First, logging on to the TD Ameritrade web site involves using the Log On button, a hyperlink. Dkt. 242 at 2; *see also* Ex. 2 at ¶¶ 125-27. Droplets admits that pressing the “enter” key to log on has the same effect as clicking the Log On button. Ex. 3 at 41:17-19; 75:10-15. Thus, logging on is selection of a hyperlink, and the interactive link cannot be a hyperlink. Dkt. 242.

Second, the client requests and retrieves the tda.js JavaScript code from the server “when a user navigates to the dock” (i.e., upon logging on). Ex. 5 at ¶ 203. However, the Log On button, username, and password fields required for log-on are not visual embodiments or graphical representations of the tda.js JavaScript code, and utilizing those fields is thus not the same as selecting tda.js.

Third, logging on to the TD Ameritrade web site cannot constitute “selection” of tda.js, because that JavaScript is not yet downloaded or executable on the client computer.

The ’745 Patent contemplates selection of a graphical representation of an interactive link, and this is not the same as user action (such as logging on to a web site) that merely causes the interactive link to download. Because tda.js does not meet the “when selected by an end user” requirement for an interactive link, the Dock lacks any interactive link. Thus, the Dock does not include all elements of Claims 1 or 26.

**d. For Search Suggest, jquery.autocomplete.ext.js is not selected by an end user**

Droplets again presents no evidence of a visual embodiment or graphical representation of the alleged interactive link. Instead, Droplets alleges that jquery.autocomplete.ext.js, the purported interactive link for Search Suggest, is “selected” in three instances: when a user (a) navigates to the page containing the search box (by logging on to the E\*TRADE website), (b) clicks on the search box, or (c) types in the search box. Ex. 5 at ¶ 97. However, these actions are neither a user “selecting” jquery.autocomplete.ext.js nor selection of an interactive link.

With respect to navigation, Droplets admits the jquery.autocomplete.ext.js JavaScript code is requested and retrieved from the server and executed by the client when the client logs on to the E\*TRADE web site using her username and password. *Id.* at ¶ 92; Ex. 3 at 98:2-21. No search functionality is involved in the logon process, nor is a selection of a graphical embodiment of the accused JavaScript involved. Instead, the logon process causes the server to load a new web page from a remote server. Ex. 8. Indeed, the jquery.autocomplete.ext.js JavaScript is not even loaded until *after* logon. Ex. 2 at ¶ 85.

With respect to clicking on and/or typing within the search box, these actions merely make use of Search Suggest. The '745 Patent is clear that the interactive link and the application are not the same thing: while the interactive link's purpose is to present or execute applications, the applications provide the functionality. *See* Ex. 10 at 8:21-34; *see also id.* at 16:41-48 ("selection of the interactive link invokes the droplet-enabled application and/or information"). Thus, none of the three actions identified by Droplets constitutes selection of an interactive link as described in the '745 Patent.

Furthermore, the jquery.autocomplete.ext.js code provides a drop down list of search suggestions. Each suggestion, however, is a hyperlink. Ex. 3 at 93:10-17 (admitting that the exemplary suggestion, "amortization," clicked in "Search Suggest.mov" is a hyperlink); *see also* Section III(A)(4), *supra*. Thus, the only aspect of the Search Suggest product that a user actually clicks on is a hyperlink, and a hyperlink cannot be the interactive link under the Court's construction.

Accordingly, the accused Search Suggest product does not include all elements of the Claims 1 or 26.

**2. The accused JavaScript cannot be the interactive link because it lacks "facilities for restoring previous operating states of the application"**

Aside from selection, interactive links must also have "facilities for restoring previous operating states of the application" under the Court's claim construction. The '745 Patent explains that operating state information relates to the particular user's use of the application:

Preferably, the operating session state information 48 includes sizes and locations of windows and controls, the position of the cursor within the application, any completed or partially completed data field values, and data fields or items of, for example, inventory selected by the user during the previous session.

Ex. 10 at 25:66-26:4. Restoring this operating state means storing the particular user's state:

[T]he method includes storing at the application server information representing a first operating state of the remotely stored application and/or information for a particular user, and in response to the selection of the graphical representation, presenting the first operating state information....

*Id.* at 5:38-45; *see also id.* at 6:50-54.

The '745 Patent teaches what it means to have “facilities for restoring previous operating states of the application” by distinguishing away from prior art. The interactive link is different from bookmarks, for example, which “do not maintain information pertaining to a previous operating state of the web site.” *Id.* at 3:47-48. The interactive link is also different from cookies because cookies expire and only store information on a computer where the first operating state was first created. *Id.* at 3:55-65. In contrast, the interactive link includes information associated with “re-establishing the second communication connection to the application server for invoking and presenting the remotely stored application and/or information associated to the interactive link...” *Id.* at 5:31-38. Indeed, the '745 Patent explains that, to use an interactive link to restore on a second computer the operating state of a user’s application, the user must first transfer the link to that second computer. *See, e.g., id.* at 25:28-55. Thus, “facilities for previous operating states of the application,” in the context of the '745 Patent, must include some customized information within the interactive link itself. Otherwise, transfer of the link from one computer to another would be unnecessary.

As explained below, none of the accused JavaScript has the required facilities for restoring previous operating states. Moreover, Search Suggest restores no state information at all.

**a. For all four representative products, the accused JavaScript is not personalized for any user and therefore cannot restore a user’s prior operating state**

It is apparent that the accused JavaScript lacks “facilities for restoring previous operating states of the application” in the context of the '745 Patent because, at any given point in time, each of the four sets of accused JavaScript code is identical<sup>4</sup> on every computer on which they run. Because each user has the same JavaScript code, the code has no facilities for restoring the state of a particular user’s application.

---

<sup>4</sup> Defendants’ accused products are updated over time, and the identified JavaScript code likewise may be modified over time. The accused products as they exist at any given point in time, however, will provide identical JavaScript code to all users.



At deposition, Defendants' expert Dr. Shamos elaborated on how the accused JavaScript does not restore custom information for different users—rather, the server returns information based on who logs on to the site:

Now, chart.js, which has been identified by [Droplets' expert] Dr. Martin as the interactive link, is absolutely identical on every computer on which it runs. There are no changes. Yours and mine are the same and everybody else's in the world.

So, chart.js contains no information about what computer it's even on. And so, if I take your copy of chart.js and I move it to my computer, it's not going to restore a previous operating state of your application. And so, therefore, the information about how to get the operating state and what the operating state may be cannot possibly be contained in chart.js, because if I take [your copy of the chart.js JavaScript code] and move it to my computer, and I log on to TD Ameritrade, I'll get my previous operating state. I'm not getting your previous operating state, to the extent there is an operating state.

Ex. 9 at 53:15-54:21. Similarly, Dr. Martin testified that none of the JavaScript code stores information about operating state. Ex. 3 at 66:8-12 ("The operating state information that we were discussing, security and date range, I don't expect to see those literally written inside the chart.js resource."); *id.* at 78:20-25 (similar testimony regarding TDA.js); *id.* at 99:16-23 (similar testimony re jquery.autocomplete.ext.js); *id.* at 118:11-19 (similar testimony for ScreenerUtility.js.package.js).

To verify that the JavaScript code itself is not customized and is not responsible for restoring state, Dr. Shamos observed that customization for Interactive Charts (e.g., last security charted) and Dock (e.g., module location and placement) is preserved even when logging on at a different computer (without the user sending himself a copy of the alleged JavaScript interactive link).<sup>5</sup> Ex. 2 at ¶¶ 50, 100, 117. Thus, the username at logon is responsible for any user customization rather than the generic JavaScript code, and the JavaScript code lacks the necessary "facilities for restoring previous operating states of the application." *Id.*

---

<sup>5</sup> As explained in Section V(A)(2)(b), *infra*, Search Suggest restores no customization at all so performing this experiment would make no sense. The declaration of Dr. Shamos does not discuss performing this particular experiment with regard to the Screener because Dr. Martin's declaration was not clear as to how Screener involves any restoration of customization at all.



It is simply not possible for the accused JavaScript code to have facilities able to restore a specific user's operating state when the code has no user-specific information associated it. The accused JavaScript code cannot be the "interactive link" as claimed.

**b. With respect to Search Suggest, no state is restored at all**

Search Suggest's functionality stands in sharp contrast to the kind of restoration described in the '745 Patent. The user enters a few letters into the search box, drop down choices appear, and the user can select one suggestion to launch a search. Next, search results are displayed corresponding to the chosen selection. *See* Section III(A)(4), *supra*. Droplets offers no evidence that a user's prior Search Suggest choices, or any prior user behavior whatsoever, have any effect on subsequent search suggestions or the search results. There are no customization options, and there is no proof of any customization at all—let alone customization restored by the alleged interactive link (the jquery.autocomplete.ex.js JavaScript). *See* Ex. 2 at ¶¶ 82-84.

Because Search Suggest does not restore any user operating state at all, the jquery.autocomplete.ext.js JavaScript code cannot be an interactive link.

**c. For Claim 26, the accused JavaScript cannot be the interactive link because the accused JavaScript cannot restore itself**

The system claimed in Claim 26 comprises in part "computer program code for presenting over said network ... applications and first information ...." Ex 10 at 33:1-34:12 (emphasis added). This "computer program code" comprises, in part, "a plurality of computer program code segments embedded with informational content...." *Id.* The presentation of this computer program code happens "in response to a selection of an interactive link." *Id.*

Droplets' infringement theory alleges, for all four representative products, that the accused JavaScript is the "plurality of computer program code segments." *See* Ex. 5 at ¶¶ 82, 136, 192, 250. Droplets also alleges that the same JavaScript is the "interactive link" for Claim 26. *Id.* at ¶¶ 71 and 52, 126 and 108, 183 and 163, and 240 and 220. In other words, Droplets alleges that the accused JavaScript code is part of the "computer program code" that is presented "in response to a selection" of the same accused JavaScript code.

Droplets’ allegations create a logical impossibility and distort the meaning of interactive link. The interactive link cannot respond to itself. Because Droplets’ allegations require a tortured reading of the claim language, they should be rejected. As such, Droplets has failed to identify any possible interactive link with respect to Claim 26.

**3. The accused JavaScript cannot be the interactive link because it is “instructions to perform on an Internet browser”**

By shifting the accused interactive link from bookmarks to JavaScript, Droplets moved from one allegation barred by disclaimer to another. This Court has previously recognized that, during the ’745 Patent’s reexamination, Droplets included a “broad disclaimer” as to “Internet shortcuts.” Dkt. 242 at 5. Specifically, Droplets made the following disclaimer: “Internet shortcuts are not graphical representations of interactive links, but are instead representations of instructions to perform on an Internet browser.” Ex. 11 at 28 (emphasis added).

Both sides’ experts agree that the accused JavaScript is a set of browser instructions.<sup>6</sup> See Ex. 2 at ¶¶ 60, 88, 110, 123 (testimony by Dr. Shamos stating that the accused JavaScript code constitutes “representations of instructions to perform on an Internet browser”); Ex. 9 at 104:2-21 (clarifying during deposition that the human readable JavaScript constitutes “representations of instructions to perform” that are translated to machine language instructions prior to execution); Ex. 3 at 64:6-9 (testimony by Dr. Martin that JavaScript can be used to instruct a web browser what to display).

This undisputed fact is problematic for Droplets: Droplets has expressly disclaimed “instructions to perform on an Internet browser” as the claimed interactive link. Also, this Court already concluded that the interactive link cannot be an “Internet shortcut,” and Droplets told the

---

<sup>6</sup> To be precise, human-readable JavaScript is “representations of instructions to perform on an Internet browser,” and the browser reads and interprets the JavaScript prior to acting on the instructions. Ex. 9 at 104:2-21.

Patent Office that an Internet shortcut is “representations of instructions to perform on an Internet browser.”<sup>7</sup>

Because the accused JavaScript is indisputably “instructions to perform on an Internet browser” and Droplets represented to the Patent Office that such instructions are not an “interactive link,” the accused JavaScript cannot be an interactive link.

**B. Under the Court’s Claim Construction, The Four Representative Products Do Not Send “Operating Environment Information”**

In addition to the insurmountable problems with Droplets’ new interactive link theory, the separate limitation of sending “information relating to the operating environment of the client” computer is also lacking in each of the four accused products. The Court construed “operating environment” information as “information about a client computer’s operating system, user interface, and hardware capabilities.” Dkt. 218 at 14 (emphasis added). Notwithstanding the conjunction requiring all three separate parameters, Droplets now contends that the “operating environment” limitation is satisfied when only two of the parameters are present (operating system and browser information) so long as the third (hardware capabilities) can be inferred. Thus, Droplets’ reading of the Court’s claim construction effectively eliminates the “hardware capabilities” requirement from the “operating environment” limitation of the ’745 Patent. Accordingly, the “operating environment” limitation is also lacking.

**1. The four representative products do not send hardware information as required by Claim 1**

Claim 1 requires “sending second information relating to the operating environment of the client computer, from the client computer to the second host computer.” Ex. 10 at 30:9-11. This Court construed “operating environment” as “information about a client computer’s operating system, user interface and hardware capabilities.” Dkt. 218 at 14 (emphasis added). The Court emphasized that the “and” is important and requires all three types of information: “[t]he connector in the specification is ‘and,’ not ‘or,’” and “information about both hardware and other

---

<sup>7</sup> If Droplets is now permitted to re-define “Internet shortcut” differently than it did in the Patent Office, it will re-capture disclaimed subject matter. The Court should not permit this.

capabilities is necessary to enable the appropriate presentation of the application or information that the interactive link retrieved.” *Id.*

Factually, the parties agree that a “user-agent header” providing some information may be sent in conjunction with each representative product. *See, e.g.*, Ex. 5 at ¶ 39. User-agent headers, however, provide “user agent” information—information about “any of the various client *programs* that initiate” a web browsing request. *Id.* at ¶ 40 (quoting RFC 2730) (emphasis added). In other words, user-agent headers provide information about software, not hardware. Thus, “user-agent header” information cannot be “operating environment” information because it does not include the required information about hardware capabilities.

Droplets identifies only one user-agent header used in connection with the four representative products. *Id.* at ¶ 41. That particular user-agent header includes this string: “User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36.” Droplets contends that the user-agent header string provides hardware information based on the presence of the “WOW64” designation in the string. But WOW64 is in parentheses following “Mozilla/5.0” (a web browser version) and is therefore a parameter referring to browser-type, not hardware. *See* Ex. 9 at 124:4-10. In particular, WOW64 is a designation coined by Microsoft and defined as “the x86 emulator that allows 32-bit Windows-based applications to run seamlessly on 64-bit Windows.” Ex. 5 at ¶ 42(c). In other words, this user-agent header uses the WOW64 designation to inform a server that it is using an emulator program that allows it to run a 32-bit browser on a 64-bit version of the Windows operating system.

Defendants’ and Droplets’ experts agree that, from the WOW64 designation, one could infer that the client is using a 64-bit version of the Windows operating system and, therefore, could further infer the client computer likely has a 64-bit processor. Ex. 3 at 133:6-25; Ex. 2 at ¶¶ 72, 75. The experts also agree that something about hardware can always be inferred from knowing what operating system is used. For example, Dr. Martin testified:

Q. Likewise, a knowledgeable person could always infer something about a computer's hardware if they knew what operating system the computer was run[ning]. Correct?

A. Yes. I think you would be able to infer something about a computer based on an expression of what its operating system was alone, yes.

Ex. 3 at 132:5-10; *see also id.* at 129:11-132:4; Ex. 9 at 121:10-123:12; Ex. 2 at ¶ 75.

But information derived from multiple levels of inference is not sufficient “information” as required by the '745 Patent claims. Because some hardware information can always be inferred from software information, if inference were sufficient, there would be no difference between requiring “information about a client computer's [1] operating system, [2] user interface and [3] hardware capabilities” and requiring only “information about a client computer's [1] operating system and [2] user interface.” The Court was clear that all three elements (operating system, user interface, and hardware) are required in light of the '745 Patent specification.

Moreover, the '745 Patent's specification teaches the importance of sending hardware capabilities as operating environment information. *See, e.g.*, Ex. 10 at 8:58-9:17 (explaining that operating environment information including hardware is important because not all client computers have the same capability). The '745 Patent does not mention inferring hardware capabilities from software information. Thus, information such as WOW64 that merely permits layered inferences about hardware capabilities is not sufficient to satisfy the “hardware capabilities” portion of “operating environment information.” *Cf. Pickholtz v. Rainbow Techs., Inc.*, 284 F.3d 1365, 1373 (Fed. Cir. 2002) (while not absolute, all claim terms are presumed to have meaning).

Because Droplets identifies only WOW64 as the “hardware information” sent for the representative products and because WOW64 only identifies software (while permitting an inference about hardware), the four representative products do not infringe Claim 1 because they lack transmission of “information relating to the operating environment of the client computer.”<sup>8</sup>

---

<sup>8</sup> Defendants believe discovery would further show that the accused products do not make any use of “WOW64” to modify any information sent to the client as required by the '745 Patent's claims. Defendants reserve all rights to raise this issue if future proceedings are necessary.

## 2. Claim 1’s “operating environment information” limitation applies equally to Claim 26

Although not explicit in its language, Claim 26 also requires sending “operating environment information” because Droplets told the Patent Office it does.<sup>9</sup> During prosecution, Droplets repeatedly distinguished the ’745 Patent from cited prior art based on the operating environment information limitation. And every time this distinction was made, Droplets insisted that the distinction applied equally to Claims 1 and 26. *Cf. Southwall Tech. Inc. v. Cardinal IG Co.*, 54 F.3d 1570 (Fed. Cir. 1995) (arguments made during prosecution regarding claim term meaning are relevant to the interpretation of that term in every patent claim).

For example, in response to rejection over prior art references, Droplets explained: “Marimba and Van Hoff do not teach or suggest ‘sending second information relating to the operating environment of the client computer, for [sic] the second computer to the second host computer’.” Ex. 11 at 44. Droplets also stated that, while Van Hoff sends client operating environment information including “platform, operating system, available memory, locale, time-zone, and client-identifier,” the prior art does not disclose basing the local presentation of the application on this information as recited by Claim 1. *Id.* at 45. Droplets then explained that these positions regarding Marimba and Van Hoff are “equally applicable to claims 17 and 26.” *Id.* at 49 (emphasis added). Droplets repeated this approach *eight times* throughout the reexamination of the ’745 Patent, each time distinguishing the prior art based on the limitations expressly within Claim 1, and then explaining how such arguments were “equally applicable” to Claim 26. *See id.* at 26, 41, 49, and 60; *see also* Ex. 12 at 26, 41, 49, 60.

Similarly, during the ’745 Patent’s original prosecution, Droplets explained that the presentational information “relates to an operating environment of client computer” and insisted this limitation also was “present in independent claim 26 as filed.” Ex. 13 at 14. Droplets’ unambiguous statements in both initial prosecution and reexamination place the same limit with

---

<sup>9</sup> Defendants did not previously raise this issue in claim construction proceedings because it is not directly connected to the meaning of any particular language of Claim 26. Rather, it is a general disclaimer issue.

regard to sending “operating environment information” for Claim 26 as is explicit in Claim 1.

Thus, as with Claim 1, the four representative products do not infringe Claim 26 because sending “information relating to the operating environment of the client computer” is lacking.

**C. Because Droplets Cannot Identify Any “Interactive Link” Or “Operating Environment” Information, There Is No Infringement**

The existence of each patent claim element in each accused product is essential to Droplets’ infringement claims. If any claim element is missing, neither direct nor indirect infringement is possible. *Johnston*, 885 F.2d at 1577-78 (holding no literal infringement unless every limitation is in accused product or process exactly); *DSU Med. Corp.*, 471 F.3d at 1303 (holding patentee must prove direct infringement “for each instance of indirect infringement”).

Here, Droplets’ claims cannot survive summary judgment because Droplets has failed to provide evidence supporting the existence of certain claim elements in the representative products. As explained above, Droplets cannot establish that any of the four representative products include, as required by Claims 1 and 26, an “interactive link” or send “information relating to the operating environment of the client computer.” In fact, the available evidence establishes that these elements are absent from the accused products. Lacking evidence, Droplets relies only on unsupported conclusions of its expert, Dr. Martin. It is “well-settled,” however, “that an expert’s unsupported conclusion on the ultimate issue of infringement is insufficient to raise a genuine issue of material fact.” *Dynacore Holdings Corp. v. U.S. Phillips Corp.*, 363 F.3d 1263, 1278 (Fed. Cir. 2004).

Summary judgment of non-infringement under the doctrine of equivalents is likewise appropriate. Dr. Martin does not address the doctrine of equivalents at all. Nor has Droplets proffered any other evidence that would support applicability of the doctrine. In the absence of “particularized testimony” or “linking evidence” to establish that differences between the representative products and Claims 1 and 26 are insubstantial, any doctrine of equivalents argument fails. *Network Commerce, Inc.*, 422 F.3d at 1363.

## **VI. CONCLUSION**

For the foregoing reasons, Defendants respectfully request that this Court grant Defendants' motion, finding that each of the four representative products does not infringe U.S. Patent No. 6,687,745.



Dated: July 30, 2014

Respectfully Submitted,

By: s/ Michael B. Levin

Michael B. Levin  
California Bar No. 172329  
mlevin@wsgr.com  
WILSON SONSINI GOODRICH &  
ROSATI  
Professional Corporation  
650 Page Mill Road  
Palo Alto, California 94304  
Telephone: 650.493.9300  
Facsimile: 650.493.6811

Brian D. Range  
Texas Bar No. 24033106  
brange@wsgr.com  
WILSON SONSINI GOODRICH &  
ROSATI  
Professional Corporation  
900 South Capital of Texas Highway  
Las Cimas IV, Fifth Floor  
Austin, Texas 78746-5546  
Telephone: 512.338.5400  
Facsimile: 512.338.5499

Jessica Leigh Margolis  
New York State Bar No. 3949732  
jmargolis@wsgr.com  
Lucy Yen  
New York State Bar No. 4871653  
lyen@wsgr.com  
WILSON SONSINI GOODRICH &  
ROSATI  
Professional Corporation  
1301 Avenue of the Americas, 40th Floor  
New York, NY 10019  
Telephone: 212.999.5800  
Facsimile: 212.999.5899

**Counsel for Defendants,  
E\*TRADE FINANCIAL  
CORPORATION, E\*TRADE  
SECURITIES, LLC, E\*TRADE BANK,**

**TD AMERITRADE HOLDING  
CORPORATION, TD AMERITRADE,  
INC., SCOTTRADE, INC. AND  
SCOTTRADE FINANCIAL SERVICES,  
INC.**

**CERTIFICATE OF SERVICE**

The undersigned hereby certifies that a true and correct copy of the above and foregoing document has been served on July 30, 2014, to all counsel of record who are deemed to have consented to electronic service via the Court's CM/ECF system per Local Civil Rule 5.2.

*s/ Michael B. Levin*

Michael B. Levin